

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of processing a schedule of a plurality of sub-transactions within a parent transaction at a computer system, the method comprising:

initiating a sub-transaction, from among the plurality of sub-transactions, into system memory of the computer system prior to occurrence of an external event on which the sub-transaction depends, the sub-transaction configured to idle until the external event occurs, the [[the]] sub-transaction including a latency attribute, the latency attribute representing an estimated wait time indicating how long the sub-transaction is expected to idle waiting for the external event to occur, the sub-transaction represented by transaction boundaries that indicate when the sub-transaction starts and ends within the parent transaction;

accessing a dynamically computed latency threshold, the dynamically computed latency threshold defining a maximum amount of time the computer system is to allow the sub-transaction to remain idle in system memory waiting for an external event to occur, the dynamically computed latency threshold computed in accordance with a latency function based on one or more system performance characteristics of the computer system;

comparing the latency attribute to the latency threshold; determining that the estimated wait time for the sub-transaction exceeds the maximum amount of time the computer system is to allow the sub-transaction to remain idle in system memory waiting for an external event to occur based on the comparison;

in response to the determination, dehydrating the schedule to persist schedule state to non-volatile storage medium, including:

recognizing the transaction boundaries of the sub-transaction;

suspending execution of the schedule;

persisting the schedule state in the non-volatile storage medium based on the transaction boundaries of the sub-transaction; and

selectively de-allocating system memory allocated to the sub-transaction after suspending execution of the schedule to free up de-allocated system memory for use by other workflow actions.

2. (Previously Presented) The method of claim 1, further comprising establishing a proxy with a naming service to associate the persisted schedule state with the occurrence of the external event.

3-4. (Canceled)

5. (Previously Presented) The method of claim 2, further comprising:
detecting the occurrence of the external event subsequent to persisting the schedule state;
reading the persisted schedule state out of the non-volatile storage medium in response to detecting the occurrence of the external event;
restoring the sub-transaction into system memory in response to detecting the occurrence of the external event; and
restoring execution of the sub-transaction based on the external event and the transaction boundaries for the sub-transaction.

6. (Previously Presented) The method of claim 5, further comprising allocating system memory for execution of the schedule prior to restoring the sub-transaction into system memory.

7. (Previously Presented) The method of claim 1, wherein the latency attribute is an adjustable latency attribute according to a variable.

8. (Previously Presented) The method of claim 7, wherein the variable is related to the actual latency for completion of the workflow action.

9. (Canceled)

10. (Original) The method of claim 1, further comprising adjusting the latency threshold based on a variable.

11-52. (Canceled)

53. (Currently Amended) A computer program product for use at a computer system, the computer program product for implementing a method of processing a schedule of a plurality of sub-transactions within a parent transaction, the computer program product comprising one or more computer storage media having stored thereon computer-executable instructions that, when executed at a processor, cause the computer system to perform the method, including the following:

- initiate a sub-transaction, from among the plurality of sub-transactions, into system memory of the computer system prior to occurrence of an external event on which the sub-transaction depends, the sub-transaction configured to idle until the external event occurs, the [[the]] sub-transaction including a latency attribute, the latency attribute representing an estimated wait time indicating how long the sub-transaction is expected to idle waiting for the external event to occur, the sub-transaction represented by transaction boundaries that indicate when the sub-transaction starts and ends within the parent transaction;

- access a dynamically computed latency threshold, the dynamically computed latency threshold defining a maximum amount of time the computer system is to allow the sub-transaction to remain idle in system memory waiting for an external event to occur, the dynamically computed latency threshold computed in accordance with a latency function based on one or more system performance characteristics of the computer system;

- compare the latency attribute to the latency threshold; determine that the estimated wait time for the sub-transaction exceeds the maximum amount of time the computer system is to allow the sub-transaction to remain idle in system memory waiting for an external event to occur based on the comparison;

- in response to the determination, dehydrating the schedule to persist schedule state to non-volatile storage medium , including:

 - recognizing the transaction boundaries of the sub-transaction;

 - suspending execution of the schedule;

 - persisting the schedule state in the non-volatile storage medium based on the transaction boundaries of the sub-transaction; and

 - selectively de-allocating system memory allocated to the sub-transaction after suspending execution of the schedule to free up de-allocated system memory for use by other workflow actions.

54. (Previously Presented) The computer program product of claim 53, further comprising computer-executable instructions that, when executed at a processor, cause the computer system to establish a proxy with a naming service to associate the persisted schedule state with the occurrence of the external event..

55. (Previously Presented) The computer program product of claim 53, further comprising computer-executable instructions that, when executed at a processor, cause the computer system to:

- detect the occurrence of the external event subsequent to persisting the schedule state;

- read the persisted schedule state out of the non-volatile storage medium in response to detecting the occurrence of the external event;

- restore the sub-transaction into system memory in response to detecting the occurrence of the external event; and

- restore execution of the sub-transaction based on the external event and the transaction boundaries for the sub-transaction.

56. (Previously Presented) The computer program product of claim 55, further comprising computer-executable instructions that, when executed at a processor, cause the computer system to allocate system memory for execution of the schedule prior to restoring the sub-transaction into system memory.

57. (Previously Presented) The computer program product of claim 53, wherein the latency attribute is an adjustable latency attribute according to a variable.

58. (Previously Presented) The computer program product of claim 57, wherein the variable is related to the actual latency for completion of the workflow action.

59. (Canceled)

60. (Previously Presented) The computer program product of claim 53, further comprising adjusting the latency threshold based on a variable.